Math

ILLIAC IV
QUARTERLY PROGRESS REPORT
October, November, and December 1969

Contract No.
USAF 30(602)-4144

ILLIAC IV Doc. No. 238

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN · URBANA, ILLINOIS

ILLIAC IV

QUARTERLY PROGRESS REPORT

October, November, and December 1969

Contract No.
USAF 30(602)-4144

Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, Illinois
61801

January 15, 1970

This work was supported in part by the Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, Illinois, and in part by the Advanced Research Projects Agency as administered by the Rome Air Development Center, under Contract No. USAF 30(602)-4144.

TABLE OF CONTENTS

REPORT SUMMARY

The Advisory Committee Meeting was held at Burroughs (Paoli, Pennsylvania), October 30-31, 1969. The purpose of the meeting was to review the status of the Hardware being constructed at Burroughs, and to review the work being done on Software at the University of Illinois. Burroughs demonstrated the PE and other subsystems. It was tentatively agreed that a future meeting be planned to give a close scrutiny to programming languages being developed (e.g., TRANQUIL, GLYPNIR, FORTRAN).

The semi-conductor memories to be produced by Fairchild for Burroughs are still behind the production schedule, due to yield problems with the memory chips. The first memory delivery to Burroughs is expected March 16, 1970 (a slippage of two months); the last (70th) memory is scheduled for delivery on June 5, 1970 (a slippage of one month).

The prototype processing element has been built--Burroughs has begun production of the PE--and most instructions have been "debugged." A minor "slow down" is expected in the twenty megahertz clock as a result of testing being performed on the prototype PE.

Software and hardware for the PEX Control Computer (PDP-9) have been completed; the computer will be shipped to Burroughs in January. The first of the University's production diagnostics for printed circuit cards and the processing element have been delivered to Burroughs.

Burroughs has submitted fixed price proposals for ILLIAC IV documentation, power supplies, and PE's. These proposals are currently being evaluated.

In December, a Project Business Office (PBO) was established for the ILLIAC IV Project. This office will deal with internal administration--budgets, building requirements--and will interface with the University Administration.

Bids for the construction of the Center for Advanced Computation building were received and approved by the University of Illinois Board of Trustees; a contractor has been selected, and construction of the building has started.

1. HARDWARE

1.1 Logic Simulation and Diagnostics

1.1.1 Logic Simulation

1.1.1.1 PE Simulator

The net lists (wire lists), generated during the previous quarter for the 37 Processing Element (PE) card types, have been maintained and updated during this period. In addition, net lists for the 5 Memory Logic Unit (MLU) card types were compiled from Burroughs logic diagrams; errors in translation and in card punching of the net lists were removed completely. All of the MLU card net lists have been released for simulation and test generation.

1.1.1.2 CU Simulator System

The Control Unit (CU) Card Logic Simulator System of programs (except the simulator Body Generator and Merge programs) have been modified, wherever necessary, to satisfy the requirements of the CU-Section System.

The modified Update program will delete or insert records into the backplane wire list and will rearrange the records into the form required by the WEDTR1 program.

The new WEDTR1 program will detect missing wires and produce an error list on disk which can then be used as input to the Update program. It is not possible to exactly update the wire list in this program; however, the program does provide a way to add many of the interface signals. Otherwise, this program is similar to the earlier WEDTR1 program, except that the disk spaces are larger--to accommodate the larger files.

The WEDTR2 program now accepts a file which is the result of partitioning a CU card into three parts: Input, Output, and Combinational parts. Partitioning is effected to increase the efficiency of

simulation for such situations as the occurrence of looped storage elements. During the pin sorting block, each pin is examined to see whether it belongs to Input, Output, Combinational or to any combination of the three partitions of the card. Due to the partitioning, some logic elements will be duplicated in the cards--this information is added to the file while processing it. WEDTR2 produces an arc list which takes into account the partitioning, and sorts the records by the package location and pin number.

The only modification in the Leveler, Reduce and Housekeep programs is the enlarging of the disk spaces to handle larger files.

1.1.2 Card Test Generation

1.1.2.1 Card Test Generator System

The Card Test Generator System was used during this period to obtain test patterns for some PE and CU cards.

Due to the higher complexity of the CU cards, slight modifications of the basic system--such as the ability to use manually generated input patterns with the aid of TESLA, merged with inputs generated by the Pattern Generator program as inputs to the simulator--have been implemented.

The necessity of generating additional information to be used by the Test Translation program for fault location purpose required the addition of a new program to the system. An initial version of this program is operational and a new version of the program with increased capability is under development.

1.1.2.2 Card Test Translation

Work in this field was concentrated in three principal areas: production of test dictionaries (including input and output patterns); automatic translation of test patterns into PEXTAP programs suitable for use with the CU Card Tester Adapter for the PEX; and automatic

translation of test patterns into programs which can be run on the Texas Instruments 561 Logic Tester which will be used to test PE cards.

Test patterns produced by the Test Generator System are converted into test dictionaries which are used for manual switch tester debugging of cards and to provide detailed information when failures are detected by any of the automatic procedures.

For each output pin of a board, failure modes, (and any indistinguishable equivalent failures) which can cause an anomalous output, are listed. In addition, cross references between failure modes and test patterns are supplied to simplify the task of fault location. Although the test dictionary production program previously existed in a primitive form, extensive revisions have been made to provide additional information to the user and to format the information in the most useful manner.

A working version of the PEXTAP translation program has been completed, and PEXTAP code generated by it has been sent to Burroughs for evaluation. Further improvements will be made during the next quarter.

Programs have been written which automatically generate valid programs for the TI-561A Automatic Printed Circuit Card Tester. These programs will be used to generate programs for testing various PE boards. They are generated from the wire list of each board and the failure test patterns applied to each input pin with the associated expected good output. The programs have been written within the constraints imposed by Burroughs on the assignment of input pins to the Texas Instruments 561 Logic Tester. The boards for which satisfactory test generation can be performed are static (i.e., no latches) logic boards with 27 input pins or less. A modified program will be written to generate the test program for those boards with more than 27 input pins. Special routines for setting and resetting latches, which, thus far, have not been written, are necessary for testing this type of board. In addition, possible problems exist if non-standard voltage level assignments are required on special boards.

1.1.2.3 Generation and Evaluation of Card Tests

All of the PE card tests generated by the Card Test Generator System have been released to Burroughs for production testing of PE cards. These tests, however, will be revised in accordance with card design modifications and improvements in the Test Generator System.

MLU Card Tests are being generated for production testing.

Significant improvements in the Test Generator System give higher resolution in locating failed components with test dictionaries. An example of the new test dictionary has been sent to Burroughs.

All of the revised PE card tests including the MLU card tests are ready to be generated whenever design modification information is available from Burroughs.

A correlation of failure modes in DIL's from a circuit analysis by Burroughs to a logical analysis by the Diagnostics group was completed for ten of the metal patterns. This correlates actual circuit failure modes with the failure modes assumed in the Card Test Generator. One class of circuit failures was found which is not presently covered in the Card Test Generator. It will be covered when analysis of all DIL types is completed.

1.1.3 PE Diagnostics

1.1.3.1 Path Test

The Detection Phase Path Tests for the prototype PE have been released to Burroughs after some errors in expected responses were removed. A few errors and failures in the prototype PE and the PE exerciser were found; some of them were located with the Detection Phase Path Test. The Detection Phase Path Test is also valid for production PE testing.

Since a masking function is required, the Location Phase Path Test should be executed with the PEX under computer control. (This masking function cannot be provided by the PEX--it must be performed by the PDP-9.)

The test dictionary of the Location Phase Path Tests will be useful to locate failures with the Detection Phase Path Tests until the PE exerciser is hooked up to the PDP-9 and the control program system in the PDP-9 supervises the Location Phase Path Test by its masking function.

1.1.3.2 Combinational Test

Re-evaluation of the combinational tests was started again in this period. Several functional tests will be released, one-by-one, after verification with the PE simulator.

1.1.3.3 Control Logic Tests

The main task in this quarter was to find a good algorithm to check sneak paths in the graphs corresponding to Boolean equations of PE control logic.

An algorithm with parallel checking capacity is used for the test generation (EQUATN/TESTGEN).

In order to take advantage of the parallel checking, two programs (EQUATN/RNDNAM2 and EQUATN/REVPTH1) were written prior to EQUATN/TESTGEN. These two programs change the formats of PGM output files, and some others, in order to fit the required formats of input files of EQUATN/TESTGEN.

The program EQUATN/TESTGEN has been written and is being debugged. This program will give the test patterns for the PE control logic. The number of tests is approximately 450. This set of tests is good, not only for detecting, but also for locating the failures in the PE control logic.

1.1.3.4 Functional Tests

In this quarter, functional tests were run on 23 test routines and the results were sent to Burroughs. In several cases, the PEXTAP source code was changed to eliminate discrepancies between actual and expected responses in the simulator portion of the tests.

The ACT program required modification to make it more general and to improve its execution efficiency. It now translates from assembled PEXTAP code to simulator input code and rearranges the input for parallel simulation at an average of 270 cards per minute.

1.1.4 PEX Computer and Supervisor System

Both the hardware and software systems necessary for interfacing the PDP-9 computer to the PEX were completed during this quarter.

The necessary components for the PEX interface arrived at the beginning of the quarter and interface construction was accomplished with a minimum of difficulty. The interface was completed at the same time as the PEX supervisor software. To permit the latter to be checked out, additional logic was temporarily added to the interface to simulate the reactions of the actual PEX. The interface and the software were then debugged as completely as possible without actually being connected to the PEX.

The minor modifications which must be made to the PEX are currently being done by Burroughs. It is anticipated that the PDP-9 will be shipped to Paoli at the beginning of the next quarter.

The PDP-9 has proven useful in other ways. Utility routines have been written which produce paper tapes for both the PEX and the TI-561 from B5500-generated magnetic tapes. This eliminates the need for extra-project computer services which were previously necessary for such conversions.

1.2 Design Automation

Within the past quarter, the layout of CU-boards was completed. Forty-five computerized layouts were completed at the University for Burroughs Corporation. With the completion of the CU-artwork, all of the detailed PC layout for ILLIAC IV is now complete.

2. SOFTWARE

2.1 Operating Systems

2.1.1 Operating System I

The operating system is in the last stages of coding and debugging that can be performed on the B5500. The design is completely frozen. After final debugging on the B6500, the effort will turn to 1) remedying deficiencies in the design and 2) accommodating the input/output provisions incorporated in TRANQUIL, GLYPNIR and ASK.

2.1.2 Operating System II

2.1.2.1 Introduction to Operating System II

The primary motivation for the SYSTEM II approach is to achieve maximum use of the ILLIAC IV array. Fundamental is that the control portion of the complete algorithm be explicitly placed on the B6500 which, as the hardware is designed, is the controller for even such intimate actions as transmissions between ILLIAC disk and PE memory. The foundations of the approach are the ILLIAC KERNEL statement and ILLIAC resource allocation declarations which will be added to a standard B6500 compiler (e.g., FORTRAN), and extensions to the B6500 Operating System (MCP) which are designed to include ILLIAC IV as a resource of the composite system.

The applications programmer merely embeds (non-nested) ILLIAC computations in his "control program" which is thereby tailored to control the set of ILLIAC computations which are embedded in it. Thus, such actions as program segment overlay and core-disk data transmissions are specified in a natural way by the applications programmer without his thinking about this facet of his program's interaction with the system. Multiprogramming between such programs, and multitasking within them is fully supported by the Burroughs MCP. Finally, an incremental

approach to ILLIAC IV is provided by the language which is central to the SYSTEM II approach since the B6500 compiler's language is a proper subset of that language.

Implementation of the MCP extensions has begun by writing them in SIMULA--a simulation language which parallels the facilities of ESPOL (the language in which the MCP is written) very closely.

2.1.2.2 The KERNEL Statement

The canonical KERNEL statement is a quadruple whose ordered four parts are:

1) PE memory declarations local to the KERNEL block;

2) Input transmissions from ILLIAC disk to PE memory all of which must be recognized as complete by the B6500 before the next step can proceed;

3) The ILLIAC computational algorithm;

4) Output transmissions from PE memory to ILLIAC disk, which must be complete before KERNEL completion notification is given to the B6500 control program.

All of the information necessary to execute a given KERNEL must be available in the B6500 at run time when control reaches the KERNEL; thus, input and output record indices and ILLIAC memory size requirements can depend on variables whose values are available in the B6500 at the time KERNEL execution is called for. However, no parameters for a given instance of a KERNEL can depend on values determined by the execution of that instance of that KERNEL.

Any combination of the constituents of a KERNEL may be elided; not all of the allowed combinations are logically desirable, however.

Three forms of KERNEL executions are possible. They are

1) ILLIAC <KERNEL quadruple>; the B6500 waits until KERNEL completion before proceeding.

2) ILLIAC PROCESS <KERNEL quadruple>; the B6500 proceeds immediately to the next statement;

simultaneous processing occurs on the ILLIAC array and the B6500.

3) ILLIAC ON <event designator> <KERNEL quadruple>; KERNEL completion causes the specified EVENT; simultaneous processing occurs on the ILLIAC array and the B6500.

2.2 Translator Writing System

During this quarter, ISL/DISK was added to the system, completing the system as outlined previously. Thus, a user may write the syntax of his language in TWINKLE, write the semantics in ISL, and generate a compiler--simply by giving everything to TWINKLE/DISK--the other necessary programs will be run in order automatically. Also, it is possible to run each step independently, if that should be desirable.

The documents describing ISL, the Translator Writing System, and TWINKLE have been produced.

An effort was begun to overhaul the parser files to make them as small and as efficient as possible. A patch to the ALGOL compiler to allow conditional voiding of cards was added. Also, some new action calls (parameterized, setting and testing a bit, incrementing or decrementing and testing a counter) will be added which will allow certain features to be done more quickly than would otherwise be possible.

A persistent problem with one of the Floyd production groups has been solved and the necessary changes to implement the solution will be made along with the above changes during the next quarter.

2.3 Compilers and Translators

2.3.1 TRANQUIL

The TRANQUIL group participated in a benchmark effort which clearly showed that the current compiler would produce code which would not execute sufficiently efficient to attract users. In the interest

of improving these figures, the TRANQUIL group has engaged in a substantial project of evaluation.

At the same time, efforts to debug the current compiler were continued, primarily in those areas not directly affected by the evaluation. The areas of set definition and operations, procedure calls, and sequential loop control have now been successfully simulated. The TRANQUIL group has also been preparing for another reduction in personnel and general reallocation of resources.

A document entitled "A TRANQUIL Programming Primer" [1] has been produced. It will be available for distribution early in 1970. It includes a good description of the TRANQUIL language and its use, comparisons with FORTRAN and ALGOL, and exercises and answers against which the reader may test his skill.

2.3.2 GLYPNIR

A review of the status of GLYPNIR was made during the last quarter of 1969. The leadership of the project was transferred from Duncan Lawrie to James L. Parker. It was determined that GLYPNIR should be given a larger exposure to users and that classes in GLYPNIR programming would be held.

The mathematical subroutine library was almost completed. A FOR statement which utilized the machine fast loop and index instruction was added. Code has been inserted to accept the "VECTOR" declaration. Writing of a user's manual to complement the education program was begun.

Further projects for expanding the language are:

Extended indexing
Extended routing
CU index variable
Define macro facility
32-bit arithmetic

2.4 Assembler

Work was begun on the macro assembler in September. By mid-November, a first version was running on the B5500. At this time, work on the assembler was suspended for the remainder of the quarter in order to investigate the feasibility of the next version of the Operating System (System II).

2.5 B5500 Operation

	No. of Jobs	Process Hours
October	10,151	287.94
November	9,603	252.18
December	9,697	93.38

The present status of the MCP is Mark X, LEVEL 00.19. Machine use has been moderate to heavy, with an average of approximately 330 jobs being run per day (throughput has been very good).

Design Automation finished their work and gave up their use of the machine on November 24, 1969.

Batch Processing is now being offered as a service to all users. A Cold Start procedure has been initiated in which user's files are no longer dumped.

3. APPLICATIONS

3.1 Numerical Analysis

3.1.1 Monte Carlo Transport

Work on Radiation Transport phenomena, using a Monte Carlo method, is being continued. The geometry being considered is two-dimensional (cylindrical symmetry) with domains consisting of 32 x 64 cells. Density is a function of space, while the absorption and scattering coefficients are functions of temperature. The process is nonlinear in the sense that the number of photons to be emitted in each cell is dependent upon the temperature, which is in turn re-evaluated after each time step according to the energy transferred to each cell.

The process has been coded into a number of subroutines (about twenty), some of which will be combined after debugging. The programs have been written in GLYPNIR and have been partially debugged using the simulator.

3.1.2 Eigenvalues

3.1.2.1 Matrix Storage Methods

3.1.2.1.1 Matrix Storage for Jacobi's Method

The modified Jacobi method for eigenvalues of symmetric matrices has been revised.

(1) A new "triangular-skew" storage scheme has been developed [2] which makes possible a more efficient moving of elements in core. The mapping of the elements a_{ij} into PE-memory is done as follows:

$$\text{for } 0 \leq i \leq \frac{n}{2} - 1, \quad j \geq i$$

$$a_{ij} \rightarrow \text{loc } i,\ \text{PE}\left[\sqrt{n}\ (j(\text{mod } \sqrt{n})) + \left\lfloor \frac{j}{\sqrt{n}} \right\rfloor + i(\sqrt{n} + 1)\right]\ (\text{mod } n)$$

and for $\frac{n}{2} \leq i \leq n - 1$

$$a_{ij} \rightarrow \text{loc } j - \frac{n}{2} + 1,\ PE\left[\sqrt{n}\ (i(\text{mod } \sqrt{n}) + 1) + \left\lfloor \frac{i - (n/2)}{\sqrt{n}} \right\rfloor + (j - \frac{n}{2})\ (\sqrt{n} + 1) + 1\right] (\text{mod } n)$$

n is the size of the matrix such that $\sqrt{n}$ is an even integer and $\lfloor x \rfloor$ is the greatest integer $\leq x$. Any n not satisfying this relationship is adjusted in a manner fully described in [2].

(2) The classical and modified Jacobi methods are described in "Eigen-Value Problems" [3]. In that document a shuffle of the first row and first column was suggested in order to subject all off-diagonal elements to the elimination process.

A major time-saving factor has been found by not subjecting the matrix to the above mentioned type of orthogonal transformation but rather to the orthogonal transformation

$$H\ A\ H^t \qquad \text{with } H = \begin{bmatrix} 0 & I_1 \\ I_2 & 0 \end{bmatrix},\ H\ H^t = I$$

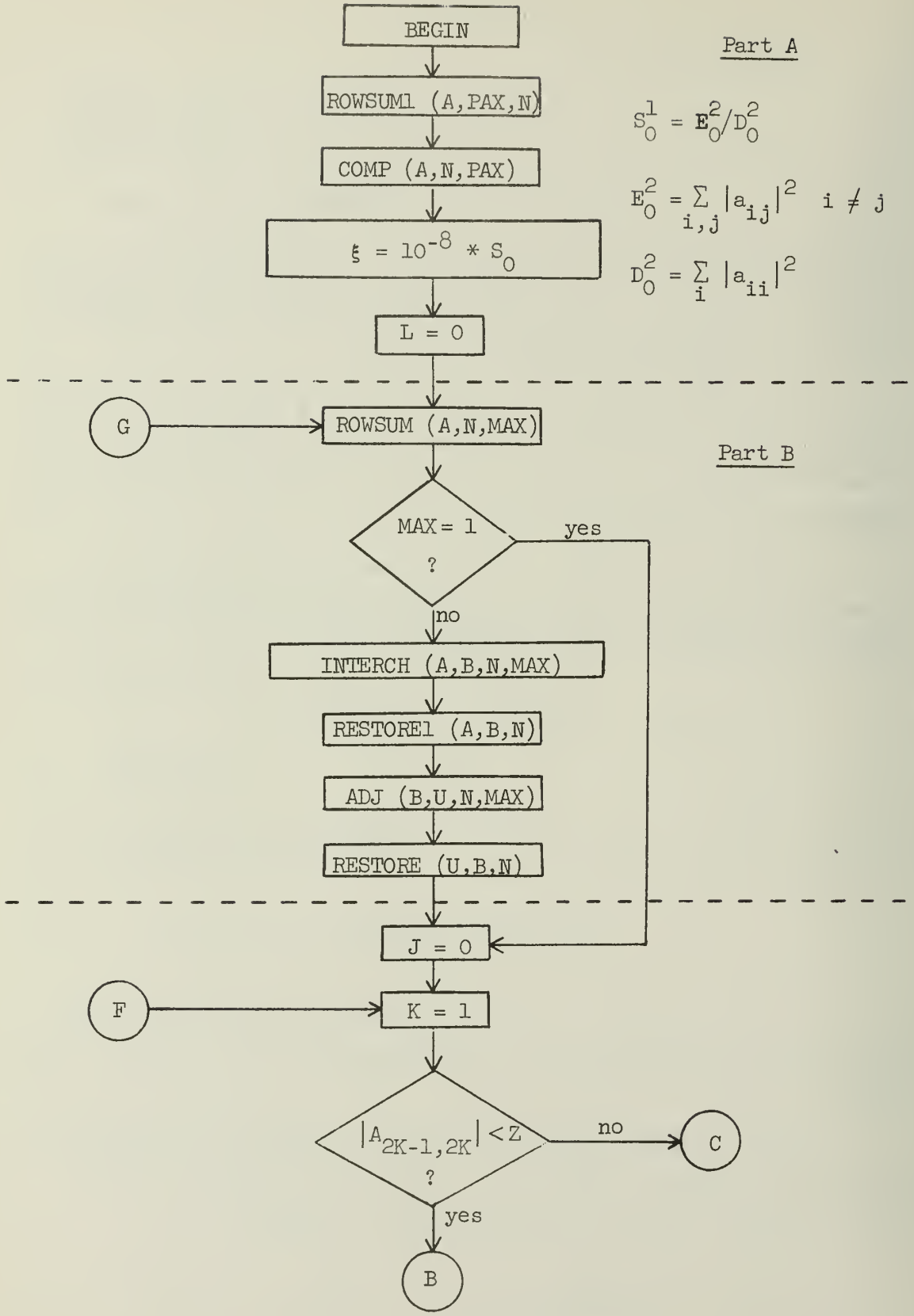
where I_1 is an $(n - m) \times (n - m)$ and I_2 is an $(m \times m)$ identity matrix. The factor m is determined by

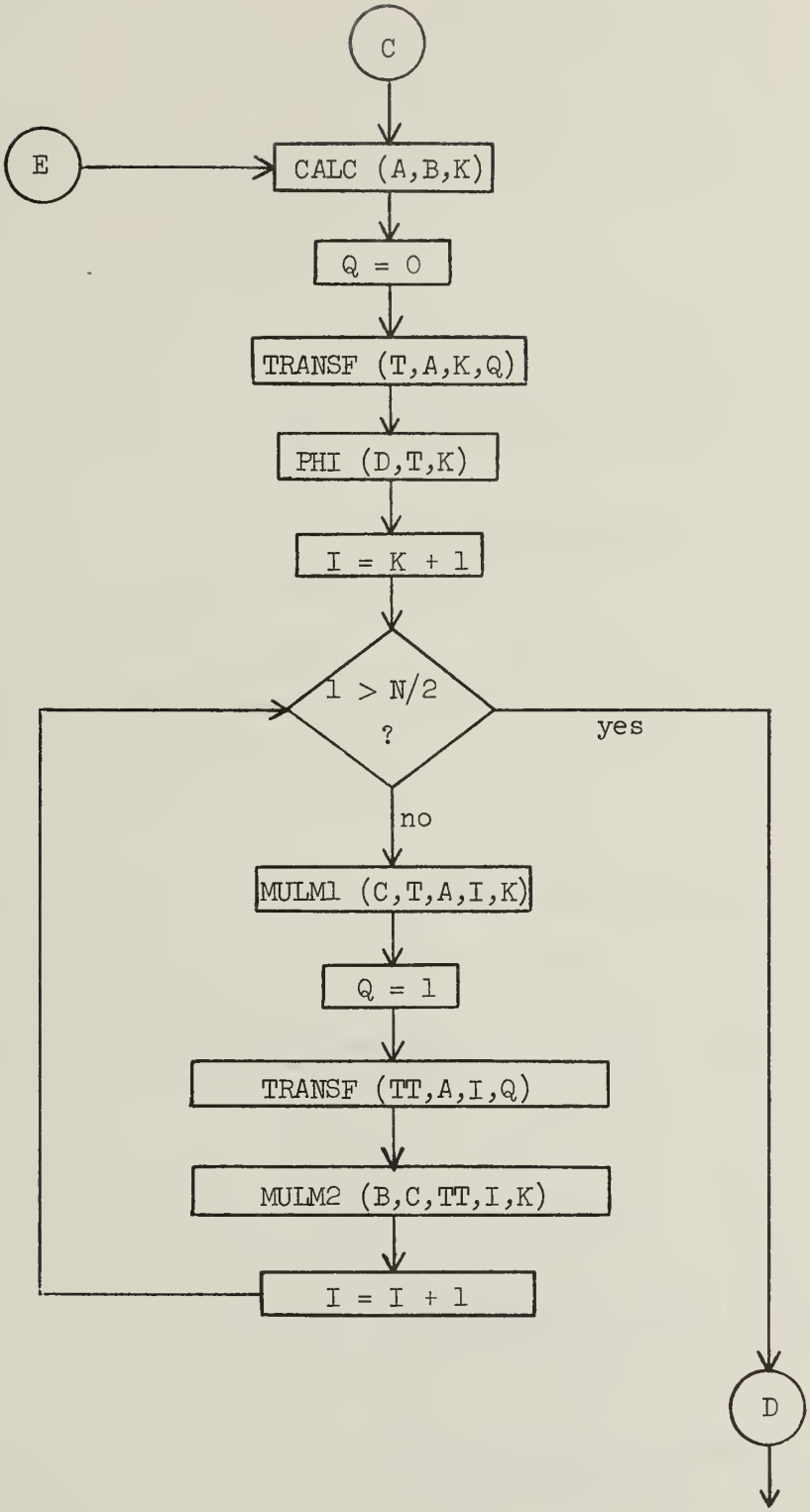
$$\max \sum_j |a_{ij}|\ (i \neq j)$$

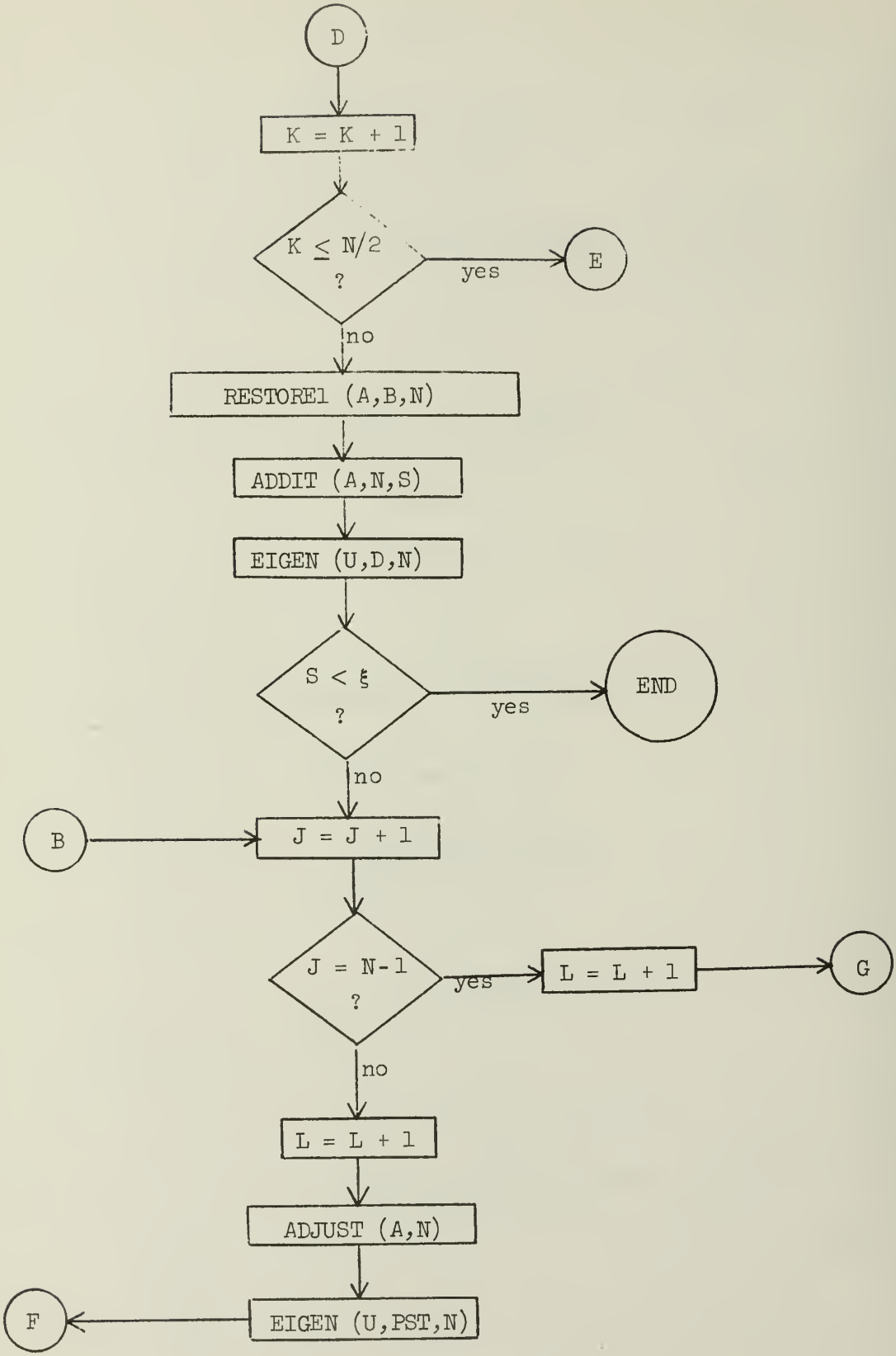
i.e., m is the row index, i, which contains the maximal value of the sum of the absolute values of the elements in this row. (See flow chart, part B.)

(3) It may be noted that Jacobi's method can be applied to the set of all symmetric matrices by subjecting the matrix A to the test of diagonal dominance according to the criterion

$$|a_{ii}| \geq \sum_{i,j} |a_{ij}| \qquad (i \neq j).$$


BEGIN
ROWSUM1 (A,PAX,N)
COMP (A,N,PAX)
ξ = 10^-8 * S_0
L = 0
G
ROWSUM (A,N,MAX)
MAX = 1 ?
yes
no
INTERCH (A,B,N,MAX)
RESTORE1 (A,B,N)
ADJ (B,U,N,MAX)
RESTORE (U,B,N)
J = 0
F
K = 1
|A_2K-1,2K| < Z ?
no
C
yes
B
Part A
S_0^1 = E_0^2/D_0^2
E_0^2 = Σ_i,j |a_ij|^2 i ≠ j
D_0^2 = Σ_i |a_ii|^2
Part B


C
E
CALC (A,B,K)
Q = 0
TRANSF (T,A,K,Q)
PHI (D,T,K)
I = K + 1
I > N/2
?
yes
no
MULM1 (C,T,A,I,K)
Q = 1
TRANSF (TT,A,I,Q)
MULM2 (B,C,TT,I,K)
I = I + 1
D


D
K = K + 1
K ≤ N/2
?
yes
E
no
RESTORE1 (A,B,N)
ADDIT (A,N,S)
EIGEN (U,D,N)
S < ξ
?
yes
END
no
B
J = J + 1
J = N-1
?
yes
L = L + 1
G
no
L = L + 1
ADJUST (A,N)
F
EIGEN (U,PST,N)

If a matrix, A, does not satisfy this criterion then a factor, b, is determined as follows:

$$b = \max \left(|a_{ii}| + \Sigma |a_{ij}|\right) \qquad (i \neq j),$$

i.e., we form a diagonally dominant matrix B = A + bI (see flow chart, part A), then

$$\Lambda = WAW^t = W(B - bI)W^t = WBW^t - bI = \Lambda' - bI$$

where $\Lambda = \text{diag}\,(\lambda_i\,(A))$. Besides a 20 percent saving in time, a gain in the accuracy of the eigenvalues and the eigenvectors has been achieved.

3.1.2.1.2 Matrix Storage for QR-Algorithm

The QR method, developed by J. Francis [4], consists of two steps: (1) reduction of a given, real matrix to almost triangular form, and (2) application of QR transformation on this matrix. A program for this method was written for the B5500 to acquire a thorough understanding of the problems involved.

The elementary elimination method and Householder's method were tested for the first step. Though Householder's method, which uses unitary transformation, is stable as far as rounding off error accumulation is concerned, the elementary elimination method requires only half the number of multiplications that Householder's method requires. The elementary elimination requires $5/6\ n^3$ multiplications in contrast to $5/3\ n^3$ multiplications for Householder's method. The result of the test indicates no significant differences between the two methods for well conditioned problems--both in accuracy of eigenvalues and in the numbers of iterations required in finding the eigenvalues from the resulting triangular matrices.

Since the QR method, as it is now, is not suitable for parallel processing, the modified QR method (QPR, October, November, and December, 1968 [5]) is being considered. This method requires

11/6 n^3 multiplications for one iteration, which corresponds to n/2 iterations in the ordinary QR method, compared with 5 n^2 for the classical method.

A study of the rate of convergence and the roundoff errors of the modified QR method is under way.

3.1.3 Pattern Matching Problem

During this quarter, a research effort was begun into the area of pattern matching. The problem is to check a source string, e.g., an English sentence, and locate a given matching pattern--a word. Such codes have been written on serial machines, and an attempt has been made to find a process to make effective use of the parallelism in ILLIAC IV. The method chosen will be referred to as the Two Phase Method. A program has been coded using ILLIAC IV assembler language, ASK/ASK II C; rough timing estimates have been obtained (about 80 times faster than the Burroughs 8500). A final stage of simulation has been reached using the ILLIAC IV simulator, SSK/SSK. More details will appear in an ILLIAC IV document.

3.2 Linear Programming

A significant portion of this quarter has been devoted to consideration of possible input and output processors for the linear programming system. Linear problems of ILLIAC IV dimensions require a sizeable amount of input data; the system should contain features which facilitate the specification (input) of these large models by the user. In addition, the conversion, pre- and post-processing of these data (outside the solution routines), consumes a substantial amount of the computer run time for the problem. Thus, the input and preprocessing facilities are an important part of a large linear programming system. The LPS preprocessor design effort will continue to interact with the development of sophisticated, flexible (user) input systems.

The investigation of the Gradient Projection method as a suitable optimization technique for large linear and nonlinear problems

was concluded this quarter. Results appear in "A Nonlinear Programming Algorithm for an Array Computer" [6], and in "Gradient Projection Method for Linear Programming" [7].

3.3 Long Codes

In order to evaluate the algorithms developed for the identification of dynamic systems, a set of programs will be developed to model a dynamic system and produce a stream of observation vectors to be manipulated by the various identification programs to be tested.

At each time step, a new system state vector is produced using the previous state vector and a predetermined transition matrix. Time-varying systems may be modeled by using a different transition matrix at each time step. Initial studies will concentrate on constant-parameter systems, where an invariant transition matrix is used. Noise vectors may be added directly to the system state vectors or to the observation vectors, which are obtained by pre-multiplying the state vectors by a predetermined observation matrix (or matrices if a time-varying system is to be modeled).

A separate noise generator program produces vectors whose elements are random variables having a Gaussian distribution with any specified covariance.

The noise generator program now has the capability to produce noise vectors having a covariance matrix equal to a scalar multiplied by an identity matrix. Also, the routines to produce the state vectors and observation vectors are essentially complete.

After the noise generator program is extended to allow more general covariance matrix specifications, and when routines, such as matrix inversion--that are used by most identification algorithms--are coded, various identification processes will be programmed and evaluated against published results.

3.4 Radar Data Processing

The Fast Fourier Transform (FFT) subroutines to handle data points (ranging from 8 to 4096) have been coded. Several parts of the subroutines have been completely debugged; however, debugging has been slow due to the inaccessibility of the B5500 computer and to the requirements of personnel working on higher priority tasks. Several new improvements have been investigated and are being incorporated into the present programs. Another routine for handling different data structures is being investigated and may be programmed as part of the routines.

Effort from this group has been applied to the improvement of 32-bit "higher" level languages for the ILLIAC IV. Changes to the GLYPNIR language, to facilitate using this language on 32-bit data structure, are being investigated. Also, the requirements for the 32-bit data handling in the layout of new languages is being investigated. The Kalman Filter Tracking Program [8] which previously was completely coded in assembly language, debugged, and executed on the simulator, has been recoded in the GLYPNIR language. The program was used to evaluate what 32-bit capability is missing in GLYPNIR and to obtain an idea of the efficiency of the higher level language on this type of problem.

3.5 Seismic Signal Processing

Two programs, autocorrelation and cross correlation, both basic requirements for a signal processing system, have been coded in ASK and are being debugged on the ILLIAC IV simulator.

Both programs are written in 32-bit and 64-bit floating point mode, using the same algorithm in each mode. Because the seismic data is real and usually contains 8-13 significant bits, the 32-bit mode allows the simultaneous processing of two data streams; whereas, the 64-bit mode provides a capability for those users whose data requires a larger word size. After debugging, the programs will be documented and added to the program library for ILLIAC IV.

3.6 Graphics

Investigations of equipment for use in the areas of interactive graphics, as well as microfilm and movie recording of graphic and alpha-numeric records, have been in progress.

Contacts with the user community have been made in order to evaluate the Project's requirements--present and future--for this type of equipment. Discussions have been held with the Project's operating system group involved in data communications in order to integrate this particular I/O channel into the system. Vendors have been approached and we have learned of the options available to us in the existing market.

At this time, we expect to understand our requirements and the options available to us on the market well enough to request a price quotation sometime in early February, with delivery of equipment anticipated in the Fall of 1970.

3.7 ILLIAC IV Education

3.7.1 CS 491-D

The graduate course in Computer Science, "Architecture, Application and Languages for a Parallel Computer," terminated this quarter. Based on the feedback from the students we will concentrate more heavily on programming languages next semester, particularly in the Assembly Language (ASK). The course will be numbered CS 491-E for the coming semester; the course outline is as follows:

I.	Hardware		6 classes
II.	Operating System		1 class
III.	Programming Languages		17 classes
		ASK	6 classes
		GLYPNIR	5 classes
		FORTRAN	4 classes
		Storage Schemes	2 classes
IV.	Applications		5 classes

3.7.2 Training Programs

A series of one-day seminars presenting an overview of ILLIAC will be offered monthly, starting in late February. The seminar will be primarily for orientation of new project personnel but will also act as an introduction of the machine to University personnel and other potential ILLIAC users. Following is a tentative outline of the seminar:

Algol and B5500	9:00 - 10:00 AM
Hardware	10:00 - noon
Instruction set (general)	1:00 - 2:00 PM
Available software	3:00 - 4:00 PM
Problem types and solutions	4:00 - 5:00 PM

In addition, more comprehensive follow-up courses will be offered in any specific area (ASK, GLYPNIR, etc.) if enough people are interested.

3.7.3 Documentation

Currently ACM (Association for Computing Machinery) category codes are being assigned to all ILLIAC IV documents. In the near future, a bibliography of all ILLIAC documents (sorted by ACM category code) will be mailed to all people currently on our mailing lists. Moreover, the FOURUM and Research Document mailing lists will be combined and all ILLIAC IV documents generated within and without the ILLIAC IV project at the University of Illinois will be assigned ACM codes and distributed through FOURUM.

3.7.4 ILLIAC IV Textbook

Work continues on the text which should be ready toward the end of the third quarter of 1970.

4. ADMINISTRATION

4.1 Administration and Services

An administrative assistant was hired in mid-December to establish a Project Business Office (PBO); thus, future reports will contain more information regarding the business operation of the project.

Bids for the construction of the Center for Advanced Computation building were received and approved by the University of Illinois Board of Trustees; a contractor has been selected, and construction of the building has started. The expected date of completion is late summer or early fall of 1970.

One of the initial steps taken by the PBO was to establish a discrete mailing address, as follows:

University of Illinois
ILLIAC IV Project
168 Engineering Research Laboratory
Urbana, Illinois 61801

At the present time, the University architect is working with the contractor in preparing bids for the electrical and mechanical systems for the Center for Advanced Computation. Burroughs is concurrently gathering similar material for bidding on the Environmental Control System of the Center.

Studies are under way to determine the need of an automatic fire extinguishing system on the computer floor and the machine room. This is being accomplished through consultation with the University architect, the University safety officer, and Burroughs Corporation.

REFERENCES

[1] Budnik, P., and Kuck, D. J., "A TRANQUIL Programming Primer," ILLIAC IV Document No. 236, DCS File No. 816, (December 10, 1969).

[2] Kuck, D., and Sameh, A., "Parallel Computation of Symmetric Matrix Eigenvalues." Unpublished.

[3] Sameh, A., and Han, L., "Eigen-Value Problems," ILLIAC IV Document No. 182, DCS Report No. 258, (April 4, 1968).

[4] Francis, J., "The QR Transformation, Parts I and II," Computer Journal, 4, 1961, pp. 265-271; 1962, pp. 332-345.

[5] ILLIAC IV Quarterly Progress Report, October, November, and December, 1968. ILLIAC IV Document No. 216, DCS Report No. 316. Urbana, Illinois: Department of Computer Science, University of Illinois, 1969.

[6] Mulvey, J. M., "A Nonlinear Programming Algorithm for an Array Computer," DCS Report No. 357. Urbana, Illinois: Department of Computer Science, University of Illinois, 1969.

[7] Yamamoto, T., and Lermit, J., "Gradient Projection Method for Linear Programming," ILLIAC IV Document No. 234, DCS File No. 820, (January 21, 1970).

[8] Stevens, J. E., Jr., "A Kalman-Filter Tracking Program for ILLIAC IV," ILLIAC IV Document No. 222, DCS File No. 804, (August 13, 1969).

THESES

Goddard, David M. "Weather Analysis on a Parallel Computer." Master's thesis, DCS Report No. 365. Urbana, Illinois: Department of Computer Science, University of Illinois, 1969.

Grothe, David M. "A Macro-Assembler for ILLIAC IV." Master's thesis, DCS Report No. 364. Urbana, Illinois: Department of Computer Science, University of Illinois, 1969.

Machado, Nelson C. "ISL--A Semantics Language for a Translator Writing System." Master's thesis, DCS Report No. 367. Urbana, Illinois: Department of Computer Science, University of Illinois, 1969.

Mulvey, John M. "A Nonlinear Programming Algorithm for an Array Computer." Master's thesis, DCS Report No. 357. Urbana, Illinois: Department of Computer Science, University of Illinois, 1969.

Pavis, Denise C. "ICL: A Control Language for ILLIAC IV." Master's thesis, DCS Report No. 356. Urbana, Illinois: Department of Computer Science, University of Illinois, 1969.

DOCUMENTS

Budnik, P., and Kuck, D. J. "A TRANQUIL Programming Primer." ILLIAC IV Document No. 236, DCS File No. 816, (December 10, 1969).

Northcote, R. S. "Some Software Considerations in Utilization of a Network of Computers." ILLIAC IV Document No. 232, DCS File No. 815, (November 21, 1969).

Yasui, T. "Double Precision Arithmetic Routine for ILLIAC IV." ILLIAC IV Document No. 228, DCS File No. 813, (October 8, 1969).

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)*
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

2a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

2b. GROUP

3. REPORT TITLE
ILLIAC IV QUARTERLY PROGRESS REPORT
October, November, December 1969

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*
October-December, 1969 - Progress Report of the ILLIAC IV Project

5. AUTHOR(S) *(First name, middle initial, last name)*

6. REPORT DATE
January 15, 1970

7a. TOTAL NO. OF PAGES
32

7b. NO. OF REFS
8

8a. CONTRACT OR GRANT NO.
USAF 30(602)-4144

b. PROJECT NO.
46-26-15-305

c.

d.

9a. ORIGINATOR'S REPORT NUMBER(S)
ILLIAC IV Document No. 238
DCS Report No. 383

9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*
RADC TR

10. DISTRIBUTION STATEMENT
Qualified requesters may obtain copies of this report from DCS.

11. SUPPLEMENTARY NOTES
NONE

12. SPONSORING MILITARY ACTIVITY
Rome Air Development Center
Griffis Air Force Base
Rome, New York 13440

13. ABSTRACT

See the Report Summary on Page 1 within the Report itself.

DD FORM 1 NOV 65 1473

UNCLASSIFIED

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Design and Construction						
Hardware Diagnostics						
Supervisory Systems						
Compilers and Generators						
ASK						
GLYPNIR						
TRANQUIL						
Numerical Analysis						
Linear Algebra						
Pattern Recognition						
Linear Programming						
Radar Data Processing						
Seismic Data Processing						
Graphics						
Texts; Handbooks						
Ordinary and Partial Differential Equations						

- 5 1972